

Dorota Roman-Jurdzińska

INFORMATYKA

1. Część informacyjna

Do egzaminu maturalnego z informatyki, który się odbył 15 maja 2012 r., przystąpiło po raz pierwszy **251** absolwentów szkół ponadgimnazjalnych (o 8% więcej maturzystów niż w roku ubiegłym), w tym **143** osób zdających wybrało ten przedmiot na poziomie podstawowym, zaś **108** na poziomie rozszerzonym (tabela 1).

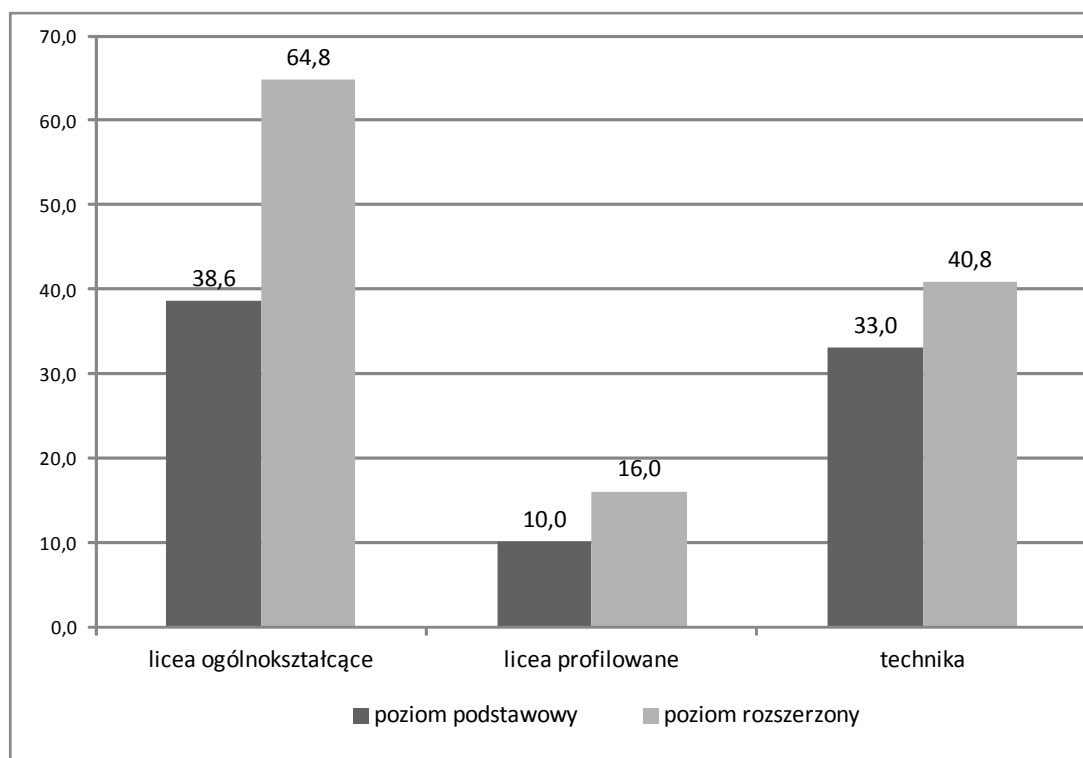
Tabela 1. Liczby uczniów na egzaminie maturalnym z informatyki – zestaw standardowy

Zdający	Liczba egzaminów		
	Ogółem	poziom podstawowy	poziom rozszerzony
<i>Okręg</i>			
LO	132	49	83
LP	2	1	1
T	117	93	24
LU			
TU			
RAZEM	251	143	108
<i>województwo dolnośląskie</i>			
LO	95	28	67
LP	2	1	1
T	69	51	18
LU			
TU			
RAZEM	166	80	86
<i>województwo opolskie</i>			
LO	37	21	16
LP			
T	48	42	6
LU			
TU			
RAZEM	85	63	22

Średni wynik procentowy z informatyki zdawanej na poziomie podstawowym wyniósł **34,7%** i jest niższy od ubiegłorocznego o 5 punktów procentowych. Dla poziomu rozszerzonego średni wynik procentowy wyniósł **59%** i jest taki sam jak w roku ubiegłym (tabela 2).

Tabela 2. Średni wynik procentowy zdających egzamin maturalny z informatyki

Typ szkoły	Średni wynik procentowy	
	poziom podstawowy	poziom rozszerzony
<i>OKE Wrocław</i>		
LO	38,6	64,8
LP	10,0	16,0
T	33,0	40,8
LU		
TU		
RAZEM	34,7	59,0
<i>województwo dolnośląskie</i>		
LO	36,8	66,7
LP	10,0	16,0
T	30,1	42,6
LU		
TU		
RAZEM	32,2	61,0
<i>województwo opolskie</i>		
LO	41,0	56,9
LP		
T	36,5	35,7
LU		
TU		
RAZEM	38,0	51,1



Wykres 1. Średnie wyniki procentowe w różnych typach szkół

Opis arkuszy egzaminacyjnych

Centralna Komisja Egzaminacyjna przygotowała dwa zestawy egzaminacyjne. Jeden dla poziomu podstawowego, drugi dla poziomu rozszerzonego. Każdy zestaw składał się z dwóch arkuszy (część teoretyczna i praktyczna), z których każdy zawierał 3 zadania o zróżnicowanej punktacji i wymagające samodzielnego sformułowania odpowiedzi. Za poprawne rozwiązanie wszystkich zadań z obu części zdający mógł uzyskać 50 punktów na każdym poziomie.

Arkusze egzaminacyjne wraz z kluczami punktowania są dostępne na stronie internetowej OKE we Wrocławiu (www.oke.wroc.pl) oraz CKE (www.cke.edu.pl).

Poziom podstawowy

Na poziomie podstawowym najtrudniejszym zadaniem było zadanie 4., sprawdzające umiejętność pisania prostego programu. Zaś bardzo łatwym zadaniem okazało się zadanie 3., które było testem wyboru i sprawdzało znajomość i rozumienie zagadnień z zakresu ogólnej wiedzy informatycznej.

Tabela 3. Łatwość zadań na poziomie podstawowym

Nr zad.	Obszar standardów	Sprawdzana umiejętność	Łatwość zadania
1.a	Wiadomości i rozumienie	Znajomość podstawowych algorytmów i technik algorytmicznych – rekurencji	0,45
1.b	Wiadomości i rozumienie	Wyodrębnianie elementów składowych algorytmu	0,88
1.c	Korzystanie z informacji	Zastosowanie podstawowych algorytmów do rozwiązywania problemów informatycznych	0,09
2.a	Wiadomości i rozumienie	Znajomość technik algorytmicznych i algorytmów	0,50
2.b	Korzystanie z informacji	Zastosowanie podstawowych algorytmów w rozwiązywaniu problemów informatycznych Analiza liczby operacji wykonywanych w algorytmie	0,20
3.a	Wiadomości i rozumienie	Znajomość sposobów reprezentowania informacji w komputerze	0,91
3.b	Wiadomości i rozumienie	Znajomość sposobów reprezentowania informacji w komputerze	0,95
3.c	Wiadomości i rozumienie	Znajomość sposobów reprezentowania informacji w komputerze	0,60
3.d	Wiadomości i rozumienie	Znajomość zasad programowania strukturalnego	0,92
3.e	Wiadomości i rozumienie	Znajomość narzędzi służących do zabezpieczania programów i danych w komputerze	0,99
4.a	Korzystanie z informacji Tworzenie informacji	Posłużenie się kompilatorem wybranego języka programowania. Sformułowanie informatycznego rozwiązania problemu przez dobór algorytmu oraz odpowiednich struktur danych i zrealizowanie go w wybranym języku programowania	0,14
4.b			0,03
4.c			0,03
5.a	Korzystanie z informacji	Dobranie właściwego programu (użytkowego lub własnoręcznie napisanego) do rozwiązywanego zadania Posłużenie się arkuszem kalkulacyjnym w celu graficznego zobrazowania informacji adekwatnie do ich charakteru	0,60
5.b			0,53
5.c			0,46
5.d			0,29

5.e		Zastosowanie odpowiedniego formatowania danych i tabeli oraz wykonanie obliczeń przy pomocy wbudowanych oraz zaprojektowanych formuł	0,16
6.a	Korzystanie z informacji	Zaprojektowanie relacyjnej bazy danych i wykorzystywanie do jej realizacji systemu bazy danych Zastosowanie metod wyszukiwania i przetwarzania informacji w relacyjnych bazach danych	0,51
6.b			0,54
6.c			0,30
6.d			0,33
6.e			0,22

Poziom rozszerzony

Na poziomie rozszerzonym najtrudniejszym zadaniem okazało się zadanie 4., które podobnie jak na poziomie podstawowym sprawdzało umiejętność implementacji algorytmów operujących na tekstach.

Tabela 4. Łatwość zadań na poziomie rozszerzonym

Nr zad.	Obszar standardów	Sprawdzana umiejętność	Łatwość zadania
1.a	Wiadomości i rozumienie	Znajomość podstawowych technik projektowania algorytmów: analiza działania funkcji rekurencyjnej dla wskazanych danych	0,61
1.b	Wiadomości i rozumienie	Znajomość i rozumienie zgodności algorytmu ze specyfikacją	0,86
1.c	Wiadomości i rozumienie	Analiza liczby operacji wykonywanych w danym algorytmie	0,54
1.d	Wiadomości i rozumienie	Znajomość podstawowych technik projektowania algorytmów: iteracja i rekurencja (algorytm znajdowania indeksu najmniejszego elementu spośród elementów o indeksach od i do n)	0,44
2.a	Wiadomości i rozumienie	Znajomość technik algorytmicznych i algorytmów (własności liczb całkowitych i naturalnych)	0,96
2.b	Wiadomości i rozumienie	Analiza liczby operacji wykonywanych w danym algorytmie	0,81
2.c	Korzystanie z informacji	Zastosowanie kolejnych etapów prowadzących do otrzymania poprawnego rozwiązania problemu: od sformułowania specyfikacji problemu po testowanie rozwiązania	0,48
3.a	Tworzenie informacji	Analiza problemu i zbioru danych	0,88
3.b	Wiadomości i rozumienie	Znajomość podstawowej terminologii związanej z sieciami komputerowymi: rodzaje sieci, protokoły	0,62
3.c	Wiadomości i rozumienie	Znajomość pozycyjnych systemów liczbowych stosowanych w informatyce	0,69
3.d	Wiadomości i rozumienie	Znajomość pozycyjnych systemów liczbowych stosowanych w informatyce	0,73
3.e	Wiadomości i rozumienie	Znajomość i opisywanie zasad etycznych i prawnych związanych z wykorzystywaniem informacji i oprogramowania	0,45
4.a	Wiadomości i rozumienie Tworzenie informacji	Znajomość technik algorytmicznych i algorytmów operujących na tekstach, zastosowanie ich w szyfrowaniu podstawieniowym Sformułowanie informatycznego rozwiązania problemu przez dobór algorytmu i odpowiednich typów oraz struktur danych i zaimplementowanie go w wybranym języku programowania (szyfrowanie i deszyfrowanie z zastosowaniem klucza)	0,39
4.b			0,35
5.a	Korzystanie z informacji	Zastosowanie odpowiedniego formatowania danych i tabeli oraz wykonanie obliczeń przy pomocy wbudowanych oraz zaprojektowanych formuł	0,68
5.b	Tworzenie informacji	Wykorzystywanie metod informatyki do rozwiązywania problemów	0,49

5.c	Tworzenie informacji	Wykorzystywanie metod informatyki do rozwiązywania problemów	0,53
5.d	Tworzenie informacji	Utworzenie dokumentu graficznego	0,54
6.a	Tworzenie informacji	Analiza problemu i zbioru danych, którego rozwiązanie wymaga zaprojektowanie i utworzenie relacyjnej bazy danych (tabel i relacji między nimi) z uwzględnieniem zawartych informacji	0,67
6.b			0,65
6.c			0,52
6.d	Korzystanie z informacji	Wyszukiwanie informacji w bazie danych stosując różne techniki (w tym zapytania) oraz zastosowanie metod optymalizujących wyszukiwanie (indeksowanie)	0,35
6.e			0,46

2. Część problemowa

Poziom podstawowy

Część I

Pierwsze dwa podpunkty **zadania 1.** dotyczyły analizy funkcji rekurencyjnej Fib. Zadaniem zdającego było prześledzenie działania podanej funkcji na konkretnych argumentach i podaniu zwracanego wyniku, a następnie narysować drzewo wywołań funkcji dla argumentu równego 6. Do poprawnego rozwiązania podpunktu a) wystarczyła jedynie znajomość definicji ciągu Fibonacciego, jednak część zdających nie potrafiła policzyć kolejnych wyrazów ciągu. W podpunkcie c) zadaniem zdających było zapisanie w postaci iteracyjnej algorytmu, który dla zdefiniowanego ciągu, dla danej wartości k wyznaczał k -ty wyraz tego ciągu. Była to najtrudniejsza część zadania. Jednym z problemów maturzystów była umiejętność pisania algorytmu iteracyjnego obliczającego wyrazy ciągu zdefiniowanego rekurencyjnie. Drugim problemem w tym zadaniu było rozróżnienie wyrazów parzystych i nieparzystych ciągu, zdający mylili „ k ” ze zmienną sterującą, tzn. sprawdzali parzystość/nieparzystość danej „ k ”, zamiast kolejnych $i=1, 2, \dots, k$. Część uczniów przepisywała algorytm z definicji bez zrozumienia i na dodatek popełniając błędy.

Przykład 1.

Algorytm:

1.	Podaj wartość k
2.	Sprawdź, czy $k=1$ lub $k=2$
3.	Jeśli $k=1$ lub $k=2$ wypisz 1
4.	Jeśli $k>2$ i k jest parzyste to zlicz sumę trzech poprzedzających go wyrazów
5.	Jeśli $k>2$ i k jest nieparzyste to k jest równy wyrazowi $k-1$
6.	Wypisz k
7.	Zamknij program

Przykład 2.

Algorytm:

$k - ty = \text{cegiła} - \text{stare} - 2 \cdot \text{cegiła}$
 $(k - ty = \text{cegiła} - \text{stare} - 2 \cdot \text{cegiła})$
 1. Podaj wartość k . $\delta = 1$
 2. jeśli $(k \geq 2 \mid 0 \leq k \bmod 2 = 0)$ (jest parzyste)
 to $k - ty = \text{cegiła} + \text{cegiła} + \text{cegiła} + \dots - 2$
 jeśli $(k \geq 2 \mid 0 \leq k \bmod 2 = 0)$ (nieparzyste)
 to $k - ty = \text{cegiła}(k - 1)$
 jeśli $(k = 1)$ to $k - ty = 1$ LUB $k - ty = 2$.
 3. wypisz $k - ty$

Zadanie 2. podobnie jak zadanie 1. polegało na analizie algorytmu dla wskazanych danych, a następnie samodzielnego sformułowania i zapisania algorytmu dla podanej specyfikacji. W zadaniu podany był algorytm zapisany w pseudokodzie, którego wynikiem działania jest liczba takich sytuacji, w których diament poprzedzający ma wyższą wartość niż diament następujący po nim. W podpunkcie a) należało przeanalizować działanie algorytmu dla czterech podanych wartości diamentów. Wielu zdających dość dobrze poradziła sobie z tą analizą, natomiast podpunkt b), w którym należało samodzielnie sformułować i zapisać algorytm, okazał się trudny (0,20). Zapisany algorytm powinien, dla podanego ciągu cen diamentów, znajdować numer diamentu o najwyższej cenie. Należy zaznaczyć, że zadanie to nie powinno nastęrczać dużych trudności, ponieważ wyszukiwanie indeksu maksimum w ciągu liczb to jedno z klasycznych zadań w podstawowym kursie algorytmiki i programowania, a jego różne warianty występowały już w arkuszach egzaminacyjnych w latach ubiegłych. Najczęstszym błędem było mylenie ceny diamentu i jego pozycji (algorytm zwracał maksymalną cenę diamentu d_i zamiast jego numeru i). Błędy występowały także w organizacji pętli sterującej.

Przykład 3.

Algorytm:

```

#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    unsigned int n; cin >> n;
    int d[n];
    for(int j=0; j<n; j++) cin >> d[j];
    int k=1, i=0, x=0;
    while(k<n)
    {
        if(d[x] > d[x+1]) i++;
        k++; x++;
    }
    cout << i;
    getch();
    return 0;
}

```

Przykład 4.

Algorytm:

1.	Wprowadź n .
2.	$j \leftarrow 0$, $k \leftarrow 1$, $m \leftarrow 1$
3.	do pólki $m, k < n$
	jeżeli większe $(i, i+1)$
	jeżeli większe (j, k) to $i \leftarrow j$
	jeżeli nie to $i \leftarrow k$
	$m \leftarrow m+1$
4.	wypisz i

Zadanie 3. było testem wyboru i sprawdzało znajomość i rozumienie zagadnień z zakresu ogólnej wiedzy informatycznej. Zadanie okazało się łatwe.

Część II

Zadania z arkusza II są rozwiązywane przy użyciu komputera. Podobnie jak w części I, każde zadanie składa się z kilku podpunktów, które dotyczą tematu przewodniego zadania. Autorzy konstruując zadania do tej części egzaminu muszą uwzględnić listę środowisk, języków programowania i programów użytkowych, z której mogą wybierać zdający egzamin maturalny z informatyki. W arkuszu II tegorocznego egzaminu maturalnego znalazły się zadania, które sprawdzały następujące umiejętności:

- rozwiązanie problemów przez dobór algorytmu oraz odpowiednich struktur danych i zrealizowanie ich w wybranym języku programowania, w tym uruchamiania i testowania samodzielnie napisanych programów komputerowych (zadanie 4.);
- rozwiązanie konkretnych problemów „z życia codziennego” za pomocą arkusza kalkulacyjnego, graficznej ilustracji danych (zadanie 5.);
- przetwarzanie danych (sortowanie, filtrowanie, łączenie, wyszukiwanie) w postaci tabel w celu wydobycia z nich konkretnych informacji (zadanie 6.).

Warto podkreślić praktyczny i interdyscyplinarny charakter zadań występujących w tym arkuszu (również w poprzednich latach) oraz fakt, że wymagana tutaj umiejętność formalnego zapisywania opisowo sformułowanych warunków sprawdza nie tylko wiedzę, ale też predyspozycje niezbędne do studiowania przedmiotów ścisłych.

Zadanie 4. to zadanie programistyczne. W jego treści zapisano wprost, aby rozwiązanie było uzyskane przy pomocy samodzielnie napisanego programu komputerowego przetwarzającego dane z pliku tekstowego (w pliku znajdowało się 1000 liczb naturalnych, mniejszych niż jeden miliard, każda liczba zapisana była w osobnym wierszu), W podpunkcie a) należało podać ile jest liczb parzystych w pliku. Aby to uzyskać wystarczyło użyć operatora mod do sprawdzania parzystości liczb oraz licznika, który zliczałby te liczby. W podpunkcie b) należało rozbić liczbę na cyfry, a następnie zsumować cyfry liczby i sprawdzić, która suma jest największa, a która najmniejsza. W ostatnim podpunkcie zadaniem zdającego było znalezienie tych liczb z pliku *cyfry.txt*, których cyfry tworzą ciąg rosnący. Wystarczyło zatem, przy rozbijaniu liczby na cyfry, porównywać dwie sąsiednie liczby.

Zadanie okazało się zadaniem bardzo trudnym dla maturzystów. Jedną z przyczyn takiej sytuacji może być problem uczniów z operacjami na plikach. Dobrym sposobem zaradzenia temu problemowi w przyszłości (oprócz położenia większego nacisku na operacje na plikach) może być wykorzystanie filtrów. Dzięki nim wystarczy napisać program działający na standardowym wejściu i wyjściu, który przy pomocy filtrów można uruchomić z danymi umieszczonymi w pliku. Inną przyczyną słabego wyniku mogą być problemy z umiejętnością wyodrębniania poszczególnych cyfr z dużej liczby oraz z używaniem w programach tekstowych typów danych.

Zadanie to miało najwyższą frakcję opuszczeń. Jeżeli podjęto próbę wykonania zadania, to najczęściej tylko podpunktu a). Wystąpiły również próby rozwiązywania tego zadania za pomocą arkusza kalkulacyjnego (zamiast napisania programu), co było niezgodne z poleceniem i zostały ocenione na zero punktów.

W **zadaniu 5.** opisane zostały zasady zmian cen akcji trzech firm A, B i C na giełdzie papierów wartościowych w kolejnych 400 dniach. Optymalnym narzędziem dla tego problemu (z punktu widzenia czasu potrzebnego na rozwiązanie) wydaje się być arkusz kalkulacyjny. Ale sprawni programiści równie dobrze rozwiązywali to zadanie przy pomocy prostego programu komputerowego. Niemniej, maturzyści korzystali przede wszystkim z arkuszy kalkulacyjnych. Najłatwiejszymi podpunktami były dwa pierwsze: a) i b). Rozwiązania tych podpunktów otrzymywało się łatwo stosując proste formuły i standardowe funkcje. Najtrudniejsze dla zdających, choć wymagające zastosowania prostych formuł i funkcji logicznych, okazały się punkty d) i e). Aby rozwiązać pierwszy z nich, należało sprawdzić, czy wszystkie zmiany cen akcji są ujemne, jeśli tak, to należało je zsumować i sprawdzić, czy ich wartość bezwzględna jest większa od 20. W podpunkcie e) należało wyznaczyć ile razy w ciągu 400 dni padła rekordowa cena akcji firmy A. Rozwiązanie polegało na zastosowaniu instrukcji warunkowej (JEŻELI) w której testem logicznym byłoby sprawdzanie, czy aktualna wartość komórki jest wyższa od maksimum ze wszystkich poprzednich wartości akcji firmy.

W **zadaniu 6.** zdający korzystali z dwóch plików tekstowych stanowiących powiązane tabele (*kandydaci*, *zgłoszenia*) relacyjnej bazy danych. Tabele były powiązane relacją „jeden do wielu” poprzez identyfikator osoby. Na zadanie składało się pięć podpunktów a) – e). Realizacja podpunktu a) wymagała uzyskania informacji z jednej tabeli kandydaci, tzn. zsumowania punktów uzyskanych przez kandydatów i filtrowania danych. Udzielenie odpowiedzi w podpunkcie b) wymagało wydobywania informacji z tabeli zgłoszenia poprzez filtrowanie i zliczenie rekordów. Realizacja podpunktu c) wymagała skorzystania z dwóch tabel, zastosowania funkcji agregujących i filtrowania. Uzyskanie prawidłowej odpowiedzi w podpunkcie d) możliwe było poprzez grupowanie i filtrowanie danych z obydwu tabel. W ostatniej części zdający powinien „wyłuskać” informacje z obydwu tabel, a następnie zastosować funkcję średnia.

W zadaniu bazodanowym dwa pierwsze podpunkty okazały się dla zdających umiarkowanie, zaś pozostałe podpunkty trudne. Naturalnym narzędziem prezentacji i analizy takich danych jest system bazodanowy pozwalający dostarczone pliki (po zaimportowaniu) potraktować jak tabele relacyjnej bazy danych. Rozwiązań, w których maturzyści korzystali z systemu bazodanowego było jednak niewiele. Powodem zapewne jest fakt, że znajomość takich narzędzi jest mniejsza niż umiejętność posługiwania się arkuszami kalkulacyjnymi, które przy odpowiedniej wprawie również umożliwiają rozwiązanie tego zadania. I właśnie arkusz kalkulacyjny był narzędziem najczęściej wykorzystywanym.

Poziom rozszerzony**Część I**

Arkusz zawierał 3 zadania i podobnie jak na poziomie podstawowym każde zadanie składało się z kilku podpunktów. Zadanie 1. i zadanie 2. miały charakter otwarty, natomiast zadanie 3. składało się z pięciu pytań testowych wyboru wielokrotnego.

Zadanie 1. podzielone zostało na cztery podpunkty. Trzy pierwsze podpunkty dotyczyły analizy podanej w treści zadania funkcji rekurencyjnej. Należy zwrócić uwagę, że rekurencja jest kluczową techniką w algorytmice, jednak nie jest łatwa koncepcyjnie. W pierwszym podpunkcie dla danej 10-elementowej tablicy należało podać wyniki wywołania funkcji F dla wskazanych argumentów. Wynikiem działania funkcji jest indeks najmniejszego elementu tablicy $a[1..n]$ z zakresu od n do i , gdzie i jest parametrem funkcji. Punkt b) jest kontynuacją analizy algorytmu z punktu a), a zdający miał w pośredni sposób podać wynik działania funkcji F . Te dwa podpunkty okazały się łatwymi dla maturzystów. W punkcie c) zdający miał podać liczbę porównań wykonywanych przez funkcję F dla argumentu $i=512$ oraz tablicy składającej się z 2012 elementów. Najtrudniejszym podpunktem okazał się ostatni punkt (łatwość 0,44). Stosunkowo niski wynik tej części zadania jest zaskoczeniem. W treści zadania podana była funkcja rekurencyjna, przez trzy pierwsze podpunkty zdający analizował tą funkcję, wyniki z tych podpunktów mogą świadczyć o tym, że rozumiał jej działanie. Polecenie w punkcie d) było proste: „Zapisz funkcję F iteracyjnie”, czyli zmienić zapis rekurencyjny na iteracyjny. Pojawiające się błędne odpowiedzi polegały na wypisywaniu wartości, a nie indeksu minimum lub na rekurencyjnym zapisie algorytmu, czyli przepisaniu funkcji z treści zadania, często z błędami.

W **zadaniu 2.** zdefiniowano tzw. liczby osiągalne. W punkcie a) zdający miał za zadanie sprawdzić, czy podane dwie liczby są osiągalne, i jeżeli tak, to podać składniki sumy k i $s(k)$. Zaś w punkcie b) miał uzasadnić zależność matematyczną. Obydwa podpunkty okazały się dla maturzystów łatwe, co świadczy o dobrej podbudowie umiejętności matematycznych reprezentowanych przez zdających.

Przykład 5.

Największa czterocyfrowa liczba: 9999
Jeżeli $n = k + s(k)$ to $k < n$
Suma cyfr największej czterocyfrowej liczby $9 + 9 + 9 + 9 = 36$
Jeżeli $k < 9999$ a $s(k)$ to suma cyfr liczby k to
 $s(k) < 36$

operacji matematycznych w systemie binarnym. Na koniec, w punkcie e), sprawdzana była wiedza na temat praw użytkownika korzystającego z oprogramowania z licencją GNU GPL. Zadanie okazało się łatwe, zdający wykazali się dobrą znajomością poruszanych w zadaniu zagadnień.

Część II

Podobnie jak na poziomie podstawowym, tegoroczny arkusz na poziomie rozszerzonym zawierał trzy zadania sprawdzające:

- umiejętność programowania (zadanie 4.);
- wykonywanie obliczeń przy pomocy wbudowanych funkcji i zaprojektowanych formuł oraz graficznej ilustracji danych w arkuszu kalkulacyjnym (zadanie 5.);
- wykonywania podsumowań, statystyk i graficznych ilustracji danych w arkuszu kalkulacyjnym (zadanie 4.).

Mimo podobieństw w strukturze arkusza na obu poziomach, zadania na poziomie rozszerzonym wymagały szerszej wiedzy i bardziej złożonych umiejętności, zgodnie z zapisami w standardach wymagań egzaminacyjnych.

Zadanie 4. miało charakter programistyczny. Wyróżniało się tym, że w treści nie wymagano wprost, aby rozwiązanie było uzyskane przy pomocy samodzielnie napisanego programu komputerowego, ale rozwiązanie tego zadania z użyciem jakiegokolwiek innego programu narzędziowego skazywało zdającego na niepowodzenie. Algorytmicznie proste zadanie należało zaimplementować w wybranym języku programowania. W podpunkcie a) zdający otrzymał plik tekstowy *tj.txt* składający się z 120 słów, które należało zaszyfrować za pomocą słów kluczy zapisanych w pliku *klucze1.txt*. Metoda szyfrowania została dokładnie opisana w treści zadania i zobrazowana przykładami. Dziesięć pierwszych i ostatnich słów w pliku miało taką samą długość, jak długość odpowiadającego mu klucza. Pozostałe słowa jawne i odpowiadające im klucze są różnej długości. Taki układ danych pozwalał sprawnie wyszukiwać błędy w warunkach pętli i odpowiednio oceniać rozwiązanie. W podpunkcie b) tego zadania należało odszyfrować zakodowane słowa w pliku *sz.txt* przy pomocy słów kluczy zawartych w pliku *klucze2.txt*. Wskaźniki łatwości dla tego zadania klasyfikują je jako zadanie trudne.

W **zadaniu 5.** należało skonstruować trójkąt Pascala składający się z 30 wierszy (zasada jego tworzenia została opisana w treści zadania) i „wyłuskać” z niego odpowiedzi do pierwszych trzech podpunktów. Zadanie można rozwiązać, pisząc odpowiedni program lub stosując formuły i funkcje (matematyczne, statystyczne i logiczne) arkusza kalkulacyjnego. W ostatniej części należało sporządzić reprezentację graficzną liczb budujących trójkąt Pascala i podzielnych przez 3 (można zastosować funkcję JEŻELI, w teście logicznym wstawić funkcję MOD, zaś jako drugi argument znak „X” albo można również zastosować formatowanie warunkowe). W pracach zdających pojawiały się błędy polegające na utworzeniu „rewersu” graficznego rozkładu liczb, czyli rozkładu liczb niepodzielnych przez 3 lub błędy w postaci braku w reprezentacji graficznej kilku końcowych wierszy. Wskaźnik łatwości klasyfikuje to zadanie w grupie zadań umiarkowanie trudnych.

Zadanie 6. to zadanie bazodanowe. Dane do zadania umieszczono w trzech plikach opisujących samochody wynajmowane od firmy leasingowej. Pliki te odpowiadają tabelom relacyjnej bazy danych. Skonstruowanie kwerend w dwóch pierwszych podpunktach okazało się dla zdających łatwe. Kwerenda w podpunkcie a) odnosiła się tylko do jednej tabeli, wystarczyło odfiltrować rekordy z samochodów leasingowanych i podsumować raty za tę usługę, a następnie to samo zrobić z samochodami wynajmowanymi. W podpunkcie b) należało wydobyć informacje z dwóch tabel i odfiltrować numery rejestracyjne firmy BARTEX. Podpunkty c) d) i e) okazały się trudniejsze dla zdających. W podpunkcie c) należało policzyć liczbę pojazdów branych w leasing przez

firmy i wyszukać maksimum. Tu pojawiały się błędy polegające na pomijaniu przez zdających informacji, że chodzi o pojazdy leasingowane, a nie o wszystkie. W podpunkcie d) zdający podawali wszystkie powiaty (zamiast tylko ziemskie) lub wyszukiwali informacje tylko w jednej tabeli tablice, bez połączenia jej relacją z tabelą *usługi*. W podpunkcie e) należało połączyć wszystkie trzy tabele i skorzystać z funkcji agregujących (średnia). Podpunkt ten charakteryzował się wysoką frakcją opuszczeń.

Podsumowanie

Podczas egzaminu maturalnego z informatyki sprawdzano umiejętności absolwentów w trzech obszarach standardów wymagań:

- znajomości i rozumienia podstawowych pojęć, metod, narzędzi i procesów związanych z informatyką;
- stosowania posiadanej wiedzy do rozwiązywania zadań teoretycznych i praktycznych;
- stosowania metod informatycznych do rozwiązywania problemów.

Arkusze egzaminacyjne z informatyki składały się z różnych rodzajów zadań: zadania otwarte oraz zamknięte, zadania wymagające umiejętności programistycznych, konstruowania i analizowania algorytmów, posługiwania się programami narzędziowymi. Można je było rozwiązać różnymi metodami, przy pomocy różnych narzędzi. Decyzja wyboru metody i programu należała do zdającego, nie została narzucona „z góry” (wyjątkiem było zadanie programistyczne na poziomie podstawowym). Różnice między arkuszami z poziomu podstawowego i rozszerzonego adekwatnie odzwierciedlają odpowiednie różnice w standardach wymagań egzaminacyjnych.

Zadania były sformułowane precyzyjnie i krótko. Najłatwiejszymi dla zdających, podobnie jak w latach poprzednich, były zadania testowe. Najtrudniejszymi były zadania z algorytmiki i programowania, szczególnie dla maturzystów, którzy przystąpili do egzaminu na poziomie podstawowym (absolwenci techników stanowili 65% zdających na tym poziomie).

Egzamin maturalny z informatyki jest egzaminem z przedmiotu ogólnokształcącego i jego kluczową częścią jest algorytmika. Zagadnienia algorytmiczne pojawiają się we wszystkich standardach egzaminacyjnych (standardy mówią zarówno o znajomości pewnych algorytmów, jak i umiejętności ich konstruowania, analizowania i stosowania). We współczesnym świecie spotykamy się ze zjawiskiem nazywanym „algorytmizacją” nauki. Coraz więcej dziedzin opisuje i charakteryzuje badane procesy i zjawiska w sposób algorytmiczny (dotyczy to nie tylko kierunków ścisłych i technicznych; sztandarowym przykładem może być np. biologia molekularna). Nie tylko akademickie, ale także korporacyjne ośrodki badawcze różnych branż potrzebują coraz więcej specjalistów potrafiących programować oraz swobodnie posługiwać się terminologią algorytmiczną. Tegoroczny egzamin maturalny (na obu poziomach) sprawdza umiejętności i wiedzę z zakresu algorytmiki w dwóch pierwszych zadaniach arkusza I oraz umiejętność konstruowania algorytmów wraz z tworzeniem w oparciu o nie programów komputerowych w pierwszym zadaniu arkusza II. Wyniki tych zadań są niższe od średnich wyników całego egzaminu, co po części wynika ze stosunkowo małego nacisku na te zagadnienia w szkołach, a także z faktu, że są to tematy stosunkowo trudne.

Drugim najważniejszym obszarem zagadnień egzaminu maturalnego z informatyki jest rozwiązywanie konkretnych problemów z zastosowaniem narzędzi technologii informacyjnej (arkuszy kalkulacyjnych, systemów baz danych i in.). Zdający muszą się tutaj wykazać nie tylko znajomością tych narzędzi, ale również umiejętnością wyrażenia celów opisanych w zadaniach w precyzyjny matematycznie i „algorytmizowany” sposób. W tegorocznym zestawach maturalnym wystąpiły po dwa zadania tego typu, w arkuszu II. Wyniki uzyskane z tych zadań na są stosunkowo dobre na obu poziomach.